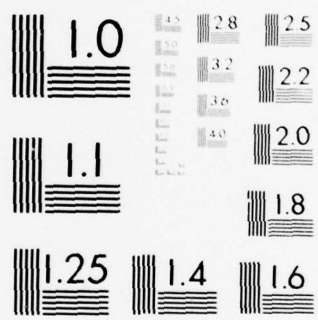
AD-A071 445 YALE UNIV NEW HAVEN CONN DEPT OF COMPUTER SCIENCE F/G 9/2
RETRIEVING INFORMATION FROM AN EPISODIC MEMORY OR WHY COMPUTERS--ETC(U)
JAN 79 R C SCHANK, J KOLODNER N00014-75-C-1111
UNCLASSIFIED RR-159 NL

| OF |

AD
A071445

END
DATE
FILMED
9-79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Retrieving Information from an
Episodic Memory
or
Why Computers' Memories Should be More
Like People's

by

Roger C. Schank and Janet Kolodner

January 1979
Research Report #159

YALE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Retrieving Information from an
Episodic Memory
or
Why Computers' Memories Should be More
Like People's

by

Roger C. Schank and Janet Kolodner

January 1979

Research Report #159

*This work was supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored under the Office of Naval Research under Contract N00014-75-C-1111.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 14 RR - 159	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 Retrieving Information from an Episodic Memory or Why Computers' Memories Should be More Like People's		5. TYPE OF REPORT & PERIOD COVERED 9 Research Rep[redacted]t
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 Roger C. Schank [redacted] Janet Kolodner		8. CONTRACT OR GRANT NUMBER(s) 15 N00014-75-C-1111
9. PERFORMING ORGANIZATION NAME AND ADDRESS Yale University - Department of Computer Science 10 Hillhouse Avenue New Haven, Connecticut 06520		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 12 35 p.
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209		12. REPORT DATE 11 Jan[redacted]79
		13. NUMBER OF PAGES 31
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research Information Systems Program Arlington, Virginia 22217		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this report is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Organization of intelligent database
Episodic Memory
Semantic Memory
Natural Language Understanding

407051 Gu

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The problem of how to organize particular experiences in a long-term memory has been largely neglected in Natural Language Understanding research. This paper addresses that problem. It shows why good memory organization is necessary for doing intelligent tasks such as story understanding and conversation. A computer memory organization modeled after human memory is proposed, as well as strategies for accessing the computer memory (again based on human information retrieval). This proposal can also be viewed as a new approach to the

next page

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

organization of intelligent databases. The CYRUS system, a computer memory model which implements this theory of memory organization, is described.

-- OFFICIAL DISTIRUBTION LIST --

Defense Documentation Center Cameron Station Alexandria, Virginia 22314	12 copies
Office of Naval Research Information Systems Program Code 437 Arlington, Virginia 22217	2 copies
Advanced Research Projects Agency Cybernetics Technology Office 1400 Wilson Boulevard Arlington, Virginia 22209	3 copies
Office of Naval Research Branch Office - Boston 495 Summer Street Boston, Massachusetts 02210	1 copy
Office of Naval Research Branch Office - Chicago 536 South Clark Street Chicago, Illinois 60615	1 copy
Office of Naval Research Branch Office - Pasadena 1030 East Green Street Pasadena, California 91106	1 copy
Mr. Steven Wong Administrative Contracting Officer New York Area Office 715 Broadway - 5th Floor New York, New York 10003	1 copy
Naval Research Laboratory Technical Information Division Code 2627 Washington, D.C. 20375	6 copies
Dr. A.L. Slafkosky Scientific Advisor Commandant of the Marine Corps Code RD-1 Washington, D.C. 20380	1 copy
Office of Naval Research Code 455 Arlington, Virginia 22217	1 copy

Office of Naval Research 1 copy
Code 458
Arlington, Virginia 22217

Naval Electronics Laboratory Center 1 copy
Advanced Software Technology Division
Code 5200
San Diego, California 92152

Mr. E.H. Gleissner 1 copy
Naval Ship Research and Development
Computation and Mathematics Department
Bethesda, Maryland 20084

Captain Grace M. Hopper 1 copy
NAICOM/MIS Planning Board
Office of the Chief of Naval Operations
Washington, D.C. 20350

Mr. Kin B. Thompson 1 copy
Technical Director
Information Systems Division
OP-91T
Office of the Chief of Naval Operations
Washington, D.C. 20350

Advanced Research Project Agency 1 copy
Information Processing Techniques
1400 Wilson Boulevard
Arlington, Virginia 22209

Professor Omar Wing 1 copy
Columbia University in the City of New York
Department of Electrical Engineering and
Computer Science
New York, New York 10027

Office of Naval Research 1 copy
Assistant Chief for Technology
Code 200
Arlington, Virginia 22217

Captain Richard L. Martin, USN 1 copy
Commanding Officer
USS Francis Marion (LPA-249)
FPO New York 09501

Major J.P. Pennell 1 copy
Headquarters, Marine Corp.
(Attn: Code CCA-40)
Washington, D.C. 20380

Retrieving Information from an Episodic Memory
or
Why Computers' Memories Should be More Like People's

Roger C. Schank and Janet Kolodner

Computer Science Department
Yale University
New Haven, Connecticut 06520

ABSTRACT

The problem of how to organize particular experiences in a long-term memory has been largely neglected in Natural Language Understanding research. This paper addresses that problem. It shows why good memory organization is necessary for doing intelligent tasks such as story understanding and conversation. A computer memory organization modeled after human memory is proposed, as well as strategies for accessing the computer memory (again based on human information retrieval). This proposal can also be viewed as a new approach to the organization of intelligent databases. The CYRUS system, a computer memory model which implements this theory of memory organization, is described.

Retrieving Information from an Episodic Memory
or
Why Computers' Memories
Should be More Like People's *

Roger Schank and Janet Kolodner

Do A.I. programs understand what they read? Much debate has taken place on this issue (e.g., Weizenbaum [1976], Dreyfus [1972]) with little agreement. One thing seems clear, however. No computer program can seriously be said to have understood anything unless it can pass a simple test of remembering what it has supposedly understood. This problem seems obvious and easily solved. We can point to programs that paraphrase or summarize. Clearly, they remember. Or do they?

Memory, after all, also involves the integration of new knowledge with old knowledge and the ability to use newly obtained information for future and as yet unspecified tasks. This is a problem that research in Natural Language Understanding has not addressed at all. In our own laboratory, we have developed programs that understand stories and summarize and answer questions about them. SAM [Cullingford, 1977] and PAM [Wilensky, 1978] use scripts and plans to read stories and answer questions about them. FRUMP [DeJong, 1979] reads newspaper stories and summarizes them. But none of these programs can apply information gained in reading one story to understand a later story. None have a long-term memory.

*This work was supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored under the Office of Naval Research under contract N00014-75-C-1111.

What does long-term memory look like? Psychologists have proposed a number of theories, none of which are completely adequate for describing the organization of episodes in memory. Quillian [1968] proposed a semantic memory that was an associative network of words used both for representing the meanings of sentences and for understanding new sentences. Anderson and Bower [1973] proposed a different kind of semantic memory for relating words together. Neither of these theories sufficiently explained how our long-term memories for experiences and events are organized. Tulving [1972] was the first to make a distinction between semantic and episodic memory, saying that semantic memory contained static knowledge about "words, concepts, and classification of concepts", while episodic memory contained information gained through personal experience. That distinction seems hazy, however, since most of our world knowledge is gained through experience. Norman [1972] pointed out that answering questions involves more than simple retrieval of information. Inference, knowledge of causality, knowledge of intent, and general world knowledge are all part of the process of retrieving information from memory. More recently, Williams [1978] has suggested that people use strategies for retrieving information from long-term memory. Neither made a proposal, however, concerning how that memory is organized. Bobrow and Norman [1975] proposed structures called schemata for relating events to each other in memory. While this seems like a reasonable device for modeling the human capability of being reminded of something, it does not address some of the other

human memory capabilities that we would like a computer memory to have.

If psychology has not told us what human memory looks like, then perhaps we must attempt to discover it ourselves. One way to do this is to look at some of the tasks a human memory does that we would want a computer to do, and design a computer program that can perform the same tasks. People do a great many different memory tasks every day in the course of understanding. We will discuss some of these below.

1. Reminding

Some events and objects remind people of previous events. A trip to the Museum of Modern Art in New York might bring to mind a previous trip to the Museum of Modern Art, a trip to New York, a trip to another museum or another museum that looks like the Museum of Modern Art, an experience of looking at modern art, and so on. It would not be unusual to hear someone say, "The main character in that play reminded me of my mother" or "That snowmobile reminds me of my motorcycle."

Should a computer be able to be reminded of things? If the only way we can manage this is to use of some "artificial reminding device", the answer is "no." But if, as seems more reasonable, the phenomenon of reminding is a natural effect of the understanding process, then the answer is "yes." We do want our programs to understand in the same way that people do. But, perhaps more importantly, the phenomonon of reminding is a

prerequisite for the process of creativity. Machines that cannot make new, undirected connections between items in memory will never be able to dream or imagine. In short, machines that cannot be reminded will not be able to think, in any serious sense of that term.

2. Relating events by time.

People are able to relate events to other events by time. A time association is made in recalling, "My daughter was born two years after we moved to California" or "I started working at Yale the year I got married."

Suppose we wanted the computer to answer the question "When did Cyrus Vance's father die?" In a discussion about Cyrus Vance, we would expect the computer to answer "When Vance was 5 years old." Like reminding, relating events by time should be part of the understanding process. People draw time relations on the fly depending upon the context of their conversation and who they are talking to. A machine that can think should be able to do that too.

3. Recognizing when statements or questions are non-sensical.

People will respond to the question "Did Nixon run for president in 1868?" a little faster than they will respond to "Did Nixon run for president in 1967?", and both questions will evoke a much quicker response than "Did Nixon run for president in 1968?" The first two questions are quickly recognized as not making sense, and therefore, no time is wasted searching memory

for an answer.

What information do people use to answer these questions? First of all, Richard Nixon is contemporary, therefore not likely to have been around in 1868. Secondly, presidential elections are in years divisible by 4, so 1967 could not have been an election year. A "thinking" computer should also be able to recognize incongruities like that.

4. Recognizing when something has not happened.

It's very easy for someone who has met few famous people to answer the question "Have you ever met a famous person in a museum?" It takes much more time for a person who has met many famous people to search all of the possibilities.

People know what they do not know. Should computers? Like recognizing incongruities, a smart computer program should recognize when it does not have the type of information being requested without having to search its entire memory.

5. Making inferences.

People can answer, "How did Cyrus Vance get to Russia?" even if they did not hear how he got there, or even if they did not know that he was there. They make the inference that he flew there because of the distance between Russia and the United States, and because Vance is a political dignitary.

We can't expect a computer to always be asked for things it explicitly has in its memory. It should be able to make simple inferences, however, from the things it does know. A program with information about Vance should be able to make the inference above when asked how Vance got to Russia.

6. Using time and place specifications.

People can calculate answers to questions such as "Who was your girlfriend in 1958?" by remembering what they were doing in 1958 or where they were living in 1958, and then remembering events at that time and who they were doing them with.

There are two issues here. First of all, a smart computer program should be able to look in the correct place in its memory to find an answer involving time and place. It should not have to look through its information about one place to find things having to do with a second place. It should be able to look in its memory around the time specified in a question without having to search through the rest of memory.

Secondly, a smart computer should be able to use information relating to time and place to calculate new lists that were not in its memory. A person who cannot name all the girlfriends he has had will nevertheless be able to calculate at least a partial list given enough time. A computer memory, like a human memory, should not contain lists of everything it knows. But, like a human memory, it should be able to search memory to find things that it does not have lists for. Time and place specifications

can be an integral part of that processing.

7. Using maps and other aids.

It is hard to name the 50 states (i.e., there is no list of them in memory), but people find it possible to name most of the states by picturing a map of the United States. Similarly, it's hard to name all the players on any particular football team, but by picturing a playing field and the positions on the field, and remembering who plays each position, it becomes an easier task.

Maps and charts, like time and place specifications, are necessary for calculating "lists" of things that are not stored as lists in memory, something a smart computer will have to do.

8. Extensive memory search.

People can search memory extensively to answer such questions as "Have you ever met a famous person?" or "Name all the museums you have been to." However, they do not always come up with complete answers.

If people don't come up with complete answers, then we can't expect computers to either. We can expect, though, that they will be able to search memory to come up with answers as extensive as people's answers. One way we evaluate a person's intelligence is by how well he remembers things. An intelligent computer should be able to search memory extensively so that it, too, can do a good job at "remembering."

But how do we design the computer analog of a human memory? How can we relate events to each other through time? How can we use time and place specifications to guide memory search? What kinds of strategies are available for searching memory?

The field of database management, which specializes in organizing large amounts of information in a computer memory, might seem the logical place to turn. Like the psychologists, however, database specialists have not addressed the problems that we are trying to solve. Conventional databases are designed to deal with only limited domains, such as employee records or inventory (see Weiderhold [1977] for an overview). They are able to keep fairly detailed records in these domains by using lists, tables, and relations, but these techniques are not sufficient for the more varied domains and tasks we want our computer memories to deal with. Databases which store full texts and use inverted files to keep track of individual words (e.g., the New York Times Database) are being used for storing large amounts of varied information, but they have no inference capabilities and can recall data only by looking for strings of words in the texts. Since all of the memory tasks we want our computer memory to do require an inference capability, conventional databases will not be very helpful in its design.

Perhaps the best way to design a computer memory is to look at specific examples of how people do what they do, and see if we can discern a human memory organization that could be used in a machine memory. Suppose we asked a friend to tell us about all

the museums he had been to. Would we expect our friend to have his memory for museum experiences organized according to a list of museums which was updated each time he visited another museum? If he had done so, it would be an easy task for him to answer this question by merely retrieving the list. There are two major problems, however, with this theory of memory organization. First, it is unreasonable to assume that people actually have lists in their heads. Second, as a practical matter, such a solution won't work in general. A memory organized in this way would have to have lists of each type of experience it knew about, and retrieval would require retrieving all the appropriate lists and doing relevant intersections and other operations on them. Such a memory would be good at finding all experiences of certain types, but it could not easily relate events to each other by time unless they were in the same list. It would similarly be incapable of making other associations between events. Ultimately, the number of lists it would need would get unwieldy.

How do people actually answer questions such as this? To find out, we did an informal experiment of asking people questions that would elicit the ways they searched their memories. A typical response to the question, "Name all the museums you have been to" went as follows:

> "Let me see. A few weeks ago, I was at the Smithsonian Institute in Washington. I saw a really good exhibit of the Centennial Exhibition there. And this past summer, I went to a bunch of museums in London -- the British Museum, some gallery whose name I can't remember, and Mme. Tussaud's Wax Museum. And I also went to some palaces there that were museums -- the Royal Pavilion in Brighton and some other palace in

> London, um,... it was called Hampton Court. What other museums have I been to? The only foreign country I've been to is England, and I just mentioned all the museums I went to there. We go to New York frequently. I've been to lots of museums there -- MOMA, the Met, the Whitney, the Guggenheim, the Brooklyn museum -- we went there once when we were visiting friends who lived in Brooklyn. There's one more I've been to. I remember we saw a good exhibit there, but I can't remember its name. I do remember that it was old and rundown, oh, and it was summer and it didn't have an air conditioner. I once went on a trip around New York state, and I went to the photography museum in Rochester and the Corning Glass Museum. I almost forgot, I've also been to a bunch of museums in Phila., Boston, and New Haven (all the places I've lived)."

We also asked people to name the most famous person they had ever met and then to tell us how they got their answers. A typical response to this question was:

> "First I thought how somebody could be famous, and politics was the first thing I thought of. Then I thought about circumstances where I could have met a famous politician. I searched political experiences I have had -- mostly political rallies I participated in and experiences campaigning for candidates. I remembered that I had met McGovern. But since you said 'most famous', I went on to think of other famous people I might have met. Next I thought of entertainers, and how I could have met them. I remembered shows I had gone to, and going back-stage. Then I thought about television programs and whether I had ever met any of those performers. Then I thought of famous scientists, and places I could have met scientists. .."

What do the answers to these two questions have in common? Both of these answers, and the answers to other questions we asked, would seem to indicate that people applied certain specific strategies to search their memories. More importantly, almost everybody we asked used the same types of strategies and broke down the problems of memory search in the same way. The human algorithm for the first question was consistently to

retrieve the most recent museum experiences first; then experiences that had been a little out of the ordinary, such as awe-inspiring experiences in museums, an inappropriate event that happened at a museum, or hearing about some museum that was out of the ordinary in some way. Next, people retrieved museums with special characteristics, such as long winding staircases, museums that used to be palaces, or museums with special types of exhibits. Finally, they retrieved places they had been and museums in those places, first naming museums in places they had visited and then in places they had lived. The algorithm for the second question involved finding the types of famous people there are and then applying strategies to search for experiences that might have led to meeting people of each type (e.g., Was I ever at a political rally? What scientific conferences have I attended?).

Let's look more closely at the museum example to see how we could get the computer to give a similar answer. The first thing humans did in answering that question was to call up their most recent and outstanding museum experiences. So, in order for the computer to use the human algorithm to retrieve museum experiences, it must associate some experiences more closely than others with museums and going to museums so that all of memory. Going to a museum basically means "doing the museum script" [Schank and Abelson, 1977]. Organizing museum experiences around the "museum script" would allow the computer to easily retrieve certain instances of the script.

After retrieving recent and outstanding experiences from memory, people searched the rest of memory for museum experiences by remembering places they had visited where they might have gone to a museum. At this point, the memory search continued as a search for places visited. People used a variety of strategies for finding places they had been. People who had not been on a large number of trips tried to remember trips they had been on. Other people, who had been on too many trips to make that feasible, followed mental maps of Europe and the United States, picking out major cities from the map and remembering their experiences in those places. The final strategy people used in finding museum experiences was to remember museums in places they had lived. The strategy rules for finding museum experiences are listed below:

1. Find museums by searching for instances of the museum script.

2. Find museum experiences (instances of the museum script) by searching for them as events within trip instances.

3. Find museums by searching memory for cities visited.

4. Find museum experiences by searching for experiences in places of residence.

5. Find trips by searching for trips to foreign countries or trips within country of residence, or by searching for foreign countries and places within country of residence and episodes associated with those places.

6. Find places by using a map of the general area.

These rules can be stored in a computer memory as features of museums, the museum script, the trip script, and each of the places. Attached to the museum script would be the information that instances of this script normally occur during instances of trips or as singular events in the place of residence. Attached to the museum stereotype would be the information that they are the location for doing the museum script and that they are usually in big cities. The trip script would hold the information that trips are to places other than the place of residence. And each place would point to memorable experiences in that place.

There would be no sense in organizing a computer memory this way unless it could be generalized to other objects and episode types. A memory set up this way would have its scripts and other stereotypes pointing to information about where in memory their instances could be found. In this way, _each stereotype would specify a set of strategies for searching memory to find its instances_. Each script and stereotype would include information about what other scripts or macro-scripts it could be a part of (eg., the museum script can be part of the trip macro-script); which of its parts are important cues for finding its instances (eg., look for places different than home to find trips); and what other memory structures can be used to find its instances (eg., go through a mental map of an area to find places). Using this organization, the following general rules for searching memory could be used:

7. To find instances of a script, search for other scripts and macro-scripts it can be part of.

8. To find objects, search for stereotypical events (such a scripts) associated with that object.

9. To find an instance of a stereotype in memory, find instances of the parts of it that are specified as important.

10. To find places, use a map.

Rule 7 is a generalization of rule 2, rule 8 is a generalization of rule 1, rule 9 is a generalization of rules 3, 4, and 5, and rule 10 is a restatement of rule 6.

In order to apply these strategies to a memory search, memory must be well-organized. A well-organized memory should be able to answer "Where did John go to law school?" without having to go through John's experiences in elementary school, his experiences at work, or his experiences with his family. We could imagine memory for experiences to be a temporally-ordered list of all of the events that were part of a person's life -- everything he did and everything he read or heard about in the course of doing things. But, this type of organization would make it difficult to retrieve any single piece of information without searching all of memory to find it. Because of the large number of events that happen within one's lifetime, straight time sequences would be hard to index into for retrieval, and therefore, not very useful for storage. Some better method of organization is obviously necessary so that the memory can be searched in a more directed way.

Consider the following knowledge we have about law school.

11. Law school constitutes a 3-year period in one's life characterized by one's being a law-school student at a certain law school.

Rule 11 says that going to law school is a time slice in a person's life characterized by being a law-school student. If all of the events that happened during this time period and that pertained to law school were grouped together in the computer memory, then only this chunk of memory would have to be searched to answer any question about law school. Questions such as "What student organizations was John involved in while he was in law school?" would be easy to answer. Consider another piece of information we know about law school:

12. Law school comes sometime after college, usually before an occupation, always before the occupation of being a lawyer.

If law school is a time slice characterized by being a law student, then college and high school are also time slices. All of these school experiences can be grouped together to form a sequence of time slices involving going to school. Suppose we were to give the computer rules 11 and 12. They would tell the program to look for law school in the school sequence after college. Once this time slice was found, the particular law school could be found by looking for those things that characterize the time slice.

An organization such as this would structure experiences temporally, but would break the time line into workable parts. It would be broken into pieces so that each piece had some unique

or partially unique characteristic (as in the example above). Each of these unique pieces or time slices is called an *era* [Kolodner 1978]. In the example above, the three years of going to law school make up a professional school era characterized by being a law-school student. Similarly, the college and high school time slices also define eras. These eras, along with any other school eras, constitute a sequence of school eras. Some of the standard ways of breaking up a personal time line are: places lived, jobs held, schools attended, and family situation. Events in a person's life having to do with his job would go in an occupational era, those having to do with his family would go in a family era, etc. Events having to do with more than one aspect of a person's life would be stored in more than one era.

Strategies for searching memory would take advantage of this memory organization. When searching for an event having to do with someone's job, only events in occupational eras would need to be checked. When searching for an event having to do with a particular job, only events in the eras corresponding to that particular job and the other eras happening at that time would have to be looked at. To implement this on a computer, each script and other stereotype would need to have information about which times in someone's life he might be involved in that type of activity, and which types of eras and era sequences it can be found in, as well as the information mentioned previously. The museum script would hold the information that residential eras can be searched to find its instances. The trip script would have the information that pleasure trips are a break from the

normal place of residence, and are also found by searching residential eras. If the political rally script for a particular person were to contain the information that he attended them most often when he was in college, then only that portion of his life would usually be searched to find political rallies. The following additional rules can be used for searching a memory organized as described above:

13. To find an instance of a stereotype in memory, look only at events in the types of eras and era sequences it could be found in.

14. To find an instance of a stereotype in memory, look only at events in the appropriate eras going on at the time when this event is most likely to have taken place.

15. To find an era, look only at events in the appropriate era sequence at the time specified relative to other eras in that sequence and in other sequences.

Applying the strategies described above will not always give complete answers. Sometimes relevant events will not be found. But people also find it hard to give complete answers to questions involving extensive memory search. People give incomplete answers when they do not apply relevant strategies. We found that people often forgot to mention museums in their home towns when naming all the museums they had been to. When people search with something specific in mind, they tend to skip over other relevant answers. When answering "Who is the most famous person you have met?", people missed finding famous entertainers while they were searching memory for politicians (even in one instance where the entertainer had introduced the person to the famous politician).

These memory organizations and strategies are not sufficient by themselves for searching memory. Besides the organization mentioned above, events must be related in a variety of other ways. It seems reasonable that people organize events around people involved in the events, countries involved, issues involved, etc. In the museum example above, some people used a mental map to find places they had visited and experiences in those places. This could not have been done if they did not have at least some events organized around those places. A computer memory following that strategy would also need to have each place point to some of the events that happened there. Similarly, each object, issue, and person should have pointers from it to some of the experiences associated with it. Organizing events in this way would allow alternative strategies of searching through events associated with each of the particular objects, countries, issues, and persons mentioned in a question.

Events must be indexed in some smart way so that they can be associated with each other as described above. Associating events with each other through a series of discrimination trees allows an event to be searched for without having to look at other events in the long event list of an era. Similar events would be near each other in memory, events that happened at the same place would be close to each other, and unusual events would be closer to the top of a tree and therefore easier to retrieve. Events could then be found by following down appropriate branches of appropriate trees. Eras would then be used for search only in cases when an event could not be found in a tree. They would

maintain their importance, however, as knowledge structures necessary for inference and time relations. The problem that remains, however, is to determine what are the relevant discriminations to be made in these trees.

How can this memory organization be used to relate events to each other by time? In order to relate events to other events by time, there must be a notion of the relative importance of events. We can imagine a person talking about a trip as occurring "two weeks before I got married." But we would find it unusual to hear that someone say that he got married "two weeks after the trip to New York when I went to the Whitney Museum." We would like a memory program to make the same types of time relations. If asked, "When did you get married?", a program should answer "in 1958, two years after I finished college" rather than "two weeks after I went to New York" (unless the current context was the trip to New York). Eras can be used for relating events by time. Events that start and end eras tend to be landmark events. Furthermore, some era sequences seem to have more importance than others. Some people tend to relate all events to occupational eras. Other people seem to use family eras for describing when events occurred.

The CYRUS system [Kolodner, 1978] is a memory model that organizes biographical information about people and uses knowledge about its organization for retrieval and automatic updating. An important aspect of the CYRUS system is that the organization of its memory represents an attempt to model memory

in people, and its retrieval and updating procedures mirror the way we believe people access their memories. CYRUS is an implementation of the ideas about memory organization and retrieval described above.

CYRUS has a question-answering module which answers questions put to it by a human user, and an updating module which automatically adds new information to memory from the UPI wire after that information has been pre-processed by FRUMP [DeJong 1979]. The CYRUS system contains information about U. S. Secretary of State Cyrus Vance, who was chosen as the model for the system since he is in the news often enough to generate a large number of news updates. A sample dialogue with CYRUS follows:*

Q1: Who is Cyrus Vance?
A1: Secretary of State of the United States.
Q2: How did he become Secretary of State?
A2: He was appointed by President Carter.
Q3: Where is he today?
A3: Egypt.
Q4: Why did he go there?
A4: He went to negotiate a compromise agreement to Egypt.
Q5: Has he met with the Israelis recently?
A5: Yes, in Washington right before he left for Egypt.

*Note: The parser and generator in CYRUS cannot yet handle all of the dialogue in these examples and the one that follow. However, the memory organization and the retrieval heuristics, which we were concentrating on in building the system, are implemented as described and can answer all the example questions posed in Conceptual Dependency.

Q6: Who was at the meeting?
A6: Moshe Dayan and some Israeli legal experts.
Q7: Will he meet with them in Israel?
A7: Yes, to discuss the compromise agreement.

Each of the memory structures and strategies mentioned above is implemented in CYRUS. The CYRUS episodic memory contains all the events in the lives of the people represented. CYRUS' stereotype memory holds each of the stereotypes that can be used for representation and all of the relevant information about each of the stereotypes.

Eras in CYRUS are characterized by the major role themes a person is involved in at each point in his life [Schank and Abelson, 1977]. In CYRUS, each person represented has an occupational sequence of eras (including school), a family sequence of eras, a social sequence of eras, and a residential sequence of eras (where each era corresponds to a place the person has lives). Each era has a corresponding role theme or place of residence. Eras contain all the events in a person's life occurring during that time span and pertaining to that role theme or place of residence. Cyrus Vance's current occupational era is characterized by his being Secretary of State; his current family era is characterized by his being a husband and father; and his current residential era is characterized by his having a place of residence in Washington, but travelling a great deal. Current events having to do with Vance's occupation go into his current occupational era; events having to do with his family life go into his current family era; and events having to do with his social life go into his current social era. Events

having to do with his place of residence go into his current residence era. Each event knows which era it lives in. When CYRUS is asked a question about a recent trip that Vance has made to Israel, it looks at events in his current occupational era, since Vance travels a lot as Secretary of State. To search for a recent party he went to, only events from his current social era are checked, unless it was a family party or had to do with his occupation, in which case events would come from either his family or occupational eras.

Events in CYRUS are represented as instantiations of scripts or macro-scripts. Each instantiation has its events attached to it through an event list. A trip to Israel in which Vance spoke to Begin and attended a state dinner would be represented as follows:

```
$TRIP actor VANCE destination ISRAEL origin USA
   events:
      $ATTEND-MEETING actor VANCE meetee BEGIN place ISRAEL
      $ATTEND-STATE-DINNER actor VANCE place ISRAEL
```

Events contained within larger episodes are organized within the episodes they are a part of. Thus, a meeting in a foreign country will always be organized as an event within the trip to that foreign country. In addition, a preliminary version of associating events with their components through the use of discrimination trees has been implemented, and events can be found by looking in the appropriate trees.

Stereotypes in the form of scripts and macro-scripts play a large role in memory retrieval. The first way to find an event asked for in a question is to search for it in the discrimination

trees associated with the components of the question. Thus, any recent or unusual events would be retrieved first. Stereotypes also specify strategies for retrieving their instances and point to eras, times, and era sequences where their instances are most likely to be found. The stereotypes CYRUS uses are scripts, macro-scripts, person stereotypes or role themes, and place stereotypes. Each script, macro-script, and role theme that is used for representation in the CYRUS episodic memory also has a stereotype in the stereotype memory. A sample script that CYRUS makes extensive use of is the attend-meeting script ($ATTEND-MEETING). Information in CYRUS about that script includes its structure, pointers to places in memory where it might be found, and recent and extraordinary events attached to it. Figure 1 shows what the attend-meeting script looks like in CYRUS:

```
scriptheader:
  ($ATTEND-MEETING actor &ACTOR attendees &MEETEES
                   topic &TOPIC place &LOC)
structure:

  &ACTOR                                    +--> &ACTOR and &MEETEES
    and    <=> *MTRANS* <-o- &TOPIC <-|
  &MEETEES                                  +--< &ACTOR and &MEETEES
          / \                 &ACTOR
           |          <-I-     and      <=> *SPEAK*
           |                  &MEETEES
         MANNER (*REPEATEDLY*)

era-sequences:   OCCUPATIONAL
macro-scripts:  $SUMMIT-CONFERENCE

recent-experiences:
  ($ATTEND-MEETING actor VANCE attendees SADAT
                   topic MID-EAST ACCORD place EGYPT
                   time (during TRIP375))

discrimination-tree:
  (TOPIC (MID-EAST-ACCORD . NODE2) (S.A.-MAJ-RULE . NODE3))
```

figure 1

Let's examine how the attend-meeting script is used by the system to answer the question, "When has Vance met with Sadat recently?" Memory must be searched for a meeting between Sadat and Vance. Information found on the attend-meeting script is used to search for an answer. CYRUS searches memory using that information. The first piece of information it uses is that these meetings occur during occupational eras. It also knows that, unless specified, events usually occur in the place of residence of one or another of the participants in an event. So CYRUS looks for meetings in the United States and Egypt. It does so by following paths in discrimination trees associated with $ATTEND-MEETING, the United States, Egypt, and Sadat. If it doesn't find any meetings by searching those trees, it uses its knowledge about places to figure out that events in Egypt had to involve a trip to Egypt, and should therefore be found by searching events within trips to Egypt. CYRUS looks for business trips to Egypt since the meeting script specifies that it is occupational.

Since all meetings do not occur in the place of residence of one or the other of the participants, this strategy is not yet complete. In the case of political VIP's, meetings often take place in another country, on neutral ground. To completely answer the question, CYRUS must use its knowledge about macro-scripts a meeting could be part of. The attend-meeting script holds the information that meetings between heads of state can be part of summit meeting episodes, and the search can continue by looking for summit meetings Vance attended where

Sadat was also a participant. CYRUS does this in much the same way it looked for regular meetings, using the additional information that summit meetings can also happen in neutral countries such as Switzerland. It can also search through his trips to those countries to look for a summit meeting. At this point in the search, if looking through appropriate discrimination trees of events has not yielded an answer, CYRUS can search through the event lists of appropriate eras to find meetings. It makes use of similar strategies to those described above, i.e., to find a meeting in Egypt, it searches through the appropriate event list for a trip to Egypt, and then through the event list of the trip for a meeting.

In order to answer this question, then, CYRUS needed to use a large amount of information. The meeting script indicated what type of eras its instances could be found in, and what macro-scripts it could also be a part of. CYRUS' general knowledge told it that the place for a meeting was probably the country of residence of one of the participants; and that in order for a person to be in a place other than his place of residence, he must take a trip there. Additionally, CYRUS needed to know that the type of a trip and its location in memory are determined by the events which happen during the trip, and should be found by searching for the appropriate type of trip. Input and output from CYRUS, including some intermediate output follow:

```
>When has Vance met with Sadat recently?

Searching for meetings between Vance and Sadat
Searching for business trips to Egypt made by Vance
Found CON562
Searching CON562 for meetings with Sadat
Found CON547
Searching CON547 for meetings with Sadat
Found one

December 15, in Egypt.
```

Some of the strategies CYRUS used in answering this question were searching directly for meetings Vance has attended, looking at experiences Vance has had with Sadat, and looking at experiences Vance has had in Egypt. Additional alternative strategies (not yet implemented) could also have been used. Alternatively, Vance's experiences associated with the Arab-Israeli conflict or the Egyptian-Israeli peace talks could have been searched (after inferring that a meeting between Vance and Sadat probably had to do with one of those issues).

In answering the question, "Has Vance ever gone sightseeing in Saudi Arabia?," information about the sightseeing script is used to guide the memory search. CYRUS first looks directly for a sightseeing experience in Saudi Arabia. If it cannot find that, CYRUS uses the place information specified in the question to infer that the sightseeing event it is looking for happened during a trip to Saudi Arabia, and it thus searches memory for trips to Saudi Arabia. It looks for pleasure trips Vance has made, since sightseeing is usually done during pleasure trips. When it finds no pleasure trips to Saudi Arabia, it must continue the search some other way. Sightseeing can also occur during

business trips, so CYRUS continues the search by looking for business trips to Saudi Arabia. A trip to Saudi Arabia in September, 1978, in which Vance went sightseeing at the oil wells in Dharan is found, and the answer is given:

```
>Has Vance ever gone sightseeing in Saudi Arabia?

Searching for pleasure trips to Saudi Arabia by
    Vance
Searching for business trips to Saudi Arabia by
    Vance
Found CON525
Searching CON525 for sightseeing experiences
Found one

Yes, he went sightseeing at the oil wells in Dharan
  last September.
```

Again, an alternative strategy could have included looking at events associated with the Arab-Israeli conflict to find a recent trip Vance made to Saudi Arabia.

Memory is searched in much the same way to answer the question, "Did Vance stay in a hotel last time he was in Saudi Arabia?" Since the place is specified as Saudi Arabia, CYRUS knows to search the events that happened during Vance's last trip to Saudi Arabia. The trip script specifies that the actor stays in a hotel unless otherwise stated. CYRUS finds Vance's last trip to Saudi Arabia as it did in the last example, and searches its events for a specification of where Vance slept. Since Vance's last trip to Saudi Arabia specified that he stayed in a guest palace, CYRUS answers accordingly:

```
>Did Vance stay in a hotel last time he was in
  Saudi Arabia?

Searching for pleasure trips to Saudi Arabia by
    Vance
Searching for business trips to Saudi Arabia by
    Vance
Found CON525
Searching CON525 for sleeping experiences
Found one

No, he slept in a guest palace there.
```

If no specification of where he slept had been found, CYRUS would have used its knowledge about trips to infer the answer "Yes".

CYRUS can use its eras and era sequences to relate events through time. If asked "When did Vance get married?", CYRUS would answer "Soon after he began to work as a lawyer." If asked, "When was he last in Paris?", it would answer "During the Vietnam peace talks, when he was an advisor to President Johnson."

CYRUS infers things not explicitly in memory by making use of expectations derived from the scripts, macro-scripts, and role themes it knows about. It uses its knowledge about enablement conditions for role themes to answer questions such as "How did Vance become Secretary of State?" In this case, the appropriate role theme tells CYRUS that Secretaries of State must be appointed by the head of state, so it uses that information to answer "He was appointed by President Carter." It can use its knowledge about trips to answer "How did Vance get to Russia?" by inferring that he took an airplane, even if this information is not explicitly in memory.

We are aiming further research at finding what other memory organizations are necessary for doing the tasks we want the computer memory to do, what other strategies are available for searching and retrieving information from memory, and how retrieval of recent and outstanding information can be more a part of the general understanding process. Although its parser and generator are not yet complete, the CYRUS memory functions can now engage in all the question-answering dialogues presented in this paper. We expect the parser and generator to be completed in the next few months. When they are complete, and after additional memory capabilities are added to CYRUS, we expect CYRUS to engage in even more sophisticated dialogues.

Earlier it was mentioned that a memory model should be able to do a lot of the memory tasks people can do. The CYRUS system applies strategies to search its episodic memory in order to give answers similar to those that a person with the same knowledge would give. It uses its organization to relate events to each other through time, and it infers information not explicitly in its memory by using the knowledge structures it knows about. In the future, we hope to explore other strategies for searching memory and more sophisticated schemes for memory organization. For example, we would like CYRUS to be able to make better use of time and place specifications on its questions in guiding memory search. We would like CYRUS to know what it does not know without extensive memory search. Giving CYRUS the ability to make more associations and to direct those associations will enable CYRUS to go off on tangents the way people do. We hope to

make CYRUS a more human-like memory, capable of making the same mistakes we see people make in information retrieval.

References

Anderson, J. and Bower, G. (1973). Human Associative Memory. Winston-Wiley, Washington, D. C.

Bobrow, D. G. and Norman, D. A. (1975). Some principles of memory schemata. In Bobrow, D. G. and Collins, A., ed. Representation and Understanding. Academic Press, Inc., New York.

Cullingford, R. (1978). Script application: computer understanding of newspaper stories. Ph. D. thesis. Research Report #116. Department of Computer Science. Yale University, New Haven, CT.

DeJong, G. F. (1979). Computerized partial understanding of connected natural language text. Ph. D. Thesis (forthcoming). Department of Computer Science. Yale University, New Haven, CT.

Dreyfus, H. L. (1972). What Computers Can't Do: A Critique of Artificial Reason. Harper & Row, New York.

Kolodner, J. L. (1978). Memory organization for natural language database inquiry. Research Report #142. Department of Computer Science. Yale University, New Haven, CT.

Quillian, M. R. (1968). Semantic Memory. In M. Minsky (ed.), Semantic Information Processing.MIT Press, Cambridge, MA.

Lehnert, W. G. (1978). The Process of Question Answering. Lawrence Erlbaum Press, Hillsdale, N.J.

Norman, D. (1972). Memory, knowledge, and the answering of questions. Center for Human Information Processing Memo CHIP-25. University of California at San Diego.

Schank, R. C. (1975). The structure of episodes in memory. In Bobrow, D. G. and Collins, A., ed. Representation and Understanding. Academic Press, Inc., New York.

Schank, R. C. and Abelson, R. P. (1977). Scripts, Plans, Goals, and Understanding. Lawrence Erlbaum Press, Hillsdale, N.J.

Tulving, E. (1972). Episodic and semantic memory. In E. Tulving & W. Donaldson (Eds.), Organization of Memory. Academic Press, New York.

Weiderhold, G. (1977). Database Design. McGraw-Hill Book Company, New York.

Weizenbaum, J. (1977). Computer Power and Human Reason: From Judgement to Calculation. Freeman, San Francisco.

Wilensky, R. (1978). Understanding goal-based stories. Ph. D. Thesis. Research Report #140. Department of Computer Science. Yale University, New Haven, CT.

Williams, M. D. (1978). The process of retrieval from very long term memory. Center for Human Information Processing Memo CHIP-75. University of California at San Diego.